

JavaScript

Les bases

Variables

```
let variable; // permet de déclarer une variable vide
               // (pour l'utiliser plus tard par exemple)

let int_var = 10; // nombre => int
let text_var = "text"; // text
let bool_var = true; // bool => true/false
let array_var = ["ratio", 10, false]; // tableau => tu peux mettre plusieurs valeurs

const MA_CONSTANTE = "xxx"; // constante, variable qu'on ne peut pas modifier
```

Afficher dans la console

```
console.log("Hello World"); // Affichera Hello World dans la console
console.error("Error"); // Comme console.log() sauf que c'est une erreur
```

Les boucles

```
// boucle "pour"
// on l'utilise quand on sait combien de fois on veut exécuter
// 3 parties séparées par ";"
// 1 - On définit la variable qui va compter le nombre de tours
// 2 - Condition, tant qu'elle est "true" (vrai) on continue
// 3 - On ajoute 1 dans i à chaque fin d'un tour de boucle
```

```

for(let i = 0; i < 10; i++) {
    //code...
}

// boucle "tant que"
// on l'utilise quand on veut lancer une boucle sans savoir
// combien de fois elle va s'exécutée
while(condition){ // Tant que la condition est "true" (vrai)
    //code...
}

```

Les fonctions

Les fonctions c'est simple: c'est comme des variable mais avec du code

```

// Fonction basique
function maFonction(){
    console.log("Bonjour");
}

// Exemple de la même fonction mais avec paramètre
function maFonctionParam(message){
    console.log(message);
}

```

Jouer avec les éléments HTML

Pour une code HTML d'exemple

```

<!DOCTYPE html>
<html>
    <head>
        <title>Cours</title>
    </head>
    <body>

```

```

    <input type="text" id="textInput">
    <button onclick="maFonction()"> Bonjour ! </button>
    <div id="daddy">
    </div>
  </body>
</html>

```

Avec ce petit bout de code on peut déjà jouer avec quelques fonctions essentielles

Récupérer un élément

Récupérer un éléments va nous permettre beaucoup de chose:

- Le modifier
- Récupérer sa valeur
- Le faire un enfants (ou lui supprimer un enfant)

```

// par CONVENTION :) quand on récup un élément de HTML on le
// Il y a plusieurs façon de récupérer les éléments HTML, par
// id ou même type de balises
// Mais d'expérience, le mieux est d'utiliser l'id

const TEXT_INPUT = document.getElementById("textInput"); // r
console.log(TEXT_INPUT); // affiche en console l'élément récup

// Très utile pour les inputs, récupérer la valeur que l'util
let textInputValue = TEXT_INPUT.value
console.log(textInputValue);

// Mettre de code html entre les balises sélectionnées
// Exemple:
const DADDY_DIV = document.getElementById("daddy");
DADDY_DIV.innerHTML = "<h1>BONSOIR PRINCESS</h1>"; // Avec in
DADDY_DIV.innerText = "<h1>BONSOIR PRINCESS</h1>" // Avec inn

```

```
// Test et regarde la différence

// Ajouter un enfant (child) à un élément
// Dans cet exemple, on ajoute une balise <p> à la div daddy
const child = document.createElement("p"); // Créer un élément
const textChild = document.createTextNode("COUCOU"); // Créer le texte
child.appendChild(textChild); // Ajouter le texte au node
DADDY_DIV.appendChild(child); // Ajouter le node à la div daddy
```

Les événements

Quand on parle d'événement, on parle de "listeners" (écouteur)

Le listener est un élément qui va attendre (ou plutôt écouter la page, d'où son nom) un certain type d'action de l'utilisateur.

Pour exemple:

- onClick ⇒ si l'utilisateur clique sur un élément (par exemple un bouton, mais tu peux le mettre sur n'importe quel genre de balise <p>)
- onSubmit ⇒ si l'utilisateur envoie un formulaire (ce qui se passe sur un formulaire)

Je t'invite à regarder de ce côté pour avoir les principaux: [JavaScript Events \(w3schools.com\)](https://www.w3schools.com/js/js_events.asp).

Il existe vraiment beaucoup de type d'événement, tu trouveras certainement toujours ton bonheur alors n'hésite surtout pas à chercher quand tu en as besoin.

Mise en place des événements / listeners

```
// 1er façon, la plus simple: sur le HTML
// Exemple avec un bouton
<button onclick="maFunction()">Bouton</button>

// 2ème façon, la plus propre: initialiser les listeners depuis le JS
// Pour un bouton aussi:
// Partie HTML
```

```
<button id="myButton">Bouton</button>
// Partie JavaScript
const MY_BUTTON = document.getElementById("myButton");
MY_BUTTON.addEventListener("click", myFunction);
```